

**REMARKS**

In the Office Action mailed June 30, 2003 the Examiner noted that claims 1-3 and 5-22 were pending, and that claims 1-3 and 5-22 have been rejected. New claims 23 through 25 have been added, thus, in view of the forgoing, claims 1-3, and 5-25 remain pending for which reconsideration is requested. No new matter has been added. The Examiner's rejections are traversed below.

**REJECTION UNDER 35 U.S.C. § 102(b)**

In the outstanding Office Action, claims 1-3 and 5-22 were rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,666,296 ("Gafter"). The rejection is traversed and reconsideration is respectfully requested.

Gafter discusses a method for symbolic evaluation of an algorithm to translate a software algorithm with a data-dependent control flow into a data flow graph from which a hardware circuit can be derived.

The present invention teaches a system that passes algorithms between nodes of a dependency graph in a graphic creation process system using a pointer to the algorithms or by passing a self-evaluating data structure.

The present invention and the prior art have two different and somewhat divergent objectives. The invention is applied to a graph intended to be repeatedly evaluated as the values within the graph as well as the connectivity of the graph itself changes. The prior art is geared towards creation of a single graph from an algorithm, which is not changed until the algorithm itself changes. The prior art is aimed at reduction of a software algorithm to hardware. The invention applies to a graphics creation process where there is a need, as is found in software as compared to hardware, for a high degree of flexibility in execution. Because of this hardware orientation, the prior art is incapable of extension to arbitrary algorithms, specifically nested function calls, whereas the present invention because of its flexibility has no such restriction.

Further, the evaluation process of the prior art specifically requires that the algorithm be fixed for analysis, whereas the invention, because of its flexibility can dynamically alter its evaluation algorithm based on the parameters, ordering, connections, and node algorithms passed through the graph. Altering of the algorithm within the prior art requires a reanalysis and

reconstruction of the equivalent or creation of a new dataflow graph. The invention can, and is designed to, because of its intentional flexibility, have its algorithms altered on the fly in response to changing data. This flexibility is provided by passing of algorithms between nodes and the execution of the algorithms at the nodes to which they have been passed based on the inputs to those nodes. The invention and the prior art will be discussed in more detail below.

The Office Action states that Gafter's symbolic evaluation of an algorithm and conversion of the algorithm into a flow graph having a plurality of nodes that each represent at least one of a plurality of statements anticipates the system disclosed by the present invention that passes functions or algorithms between dependency nodes. According to Gafter, algorithms are converted into a control flow graph ("CFG"), and are then assigned symbolic values in succession according to the control flow. See independent claims 1, 7, 13, and column 1, lines 55-67. For example, both arms of the IF statement of the software code in column 3, lines 40-50 in Gafter are translated and are accounted for in the CFG shown in FIG. 3, which is a static representation of the program in which all alternatives of the control flow are represented. Accordingly, since the CFG in Gafter is based on the algorithm, the CFG does not change unless the algorithm changes. See FIGS. 3-5 and corresponding text.

In contrast, according to the present invention, an algorithm is not translated into a CFG but is instead passed from one dependency node to another (See claims 1, and 17-22), each node having its own control structure and internal evaluation method. See page 3, lines 23-27; page 6, lines 1-11. For example, for the function shown in FIG. 2 of the present invention, the function  $f(d)=e$  itself is passed from source node 6 to destination node 8. See page 5, lines 1-9, and FIG. 2 in the specification of the present invention. As such, when a different input is entered into source node 6, destination node calls on the function  $f(d)=e$  to obtain the respective result without being burdened with the overhead of traversing the two nodes multiple times. See page 5, line 1-9, and FIG. 2 of the present invention. Assuming for argument sake that the Gafter method was to be applied to a function similar to the one illustrated in FIG. 2 of the present invention, then a representation of each variation resulting upon entry of different input data would be required. This would be very inefficient compared to the present invention. Additionally, the present invention provides a solution to this Gafter problem of having to create different nodes for each variation by "passing a pointer to the algorithm" itself and reexecuting the algorithm via "the second dependency node" each time new input data is entered. See FIG. 2 and corresponding text; independent claim 1, 17, and 19-22.

The Examiner further alleges that execution of an algorithm as part of an evaluation of the dependency node in the present invention is anticipated by the symbolical evaluation of each node in Gafter. The evaluation process in Gafter involves assuming each possible value of the input datum for each branch, and assigning a symbolic value to each of the variables. See column 2, lines 15-26. Accordingly, Gafter's evaluation process involves the use of shadow symbol tables constructed for each outcome of the algorithm. See column 5, lines 21-45. For example, for the IF statement mentioned above, two instances of the shadow symbol table are constructed and are later merged to create conditional expressions where any corresponding expression in the tables differ. See column 5, lines 45-60. Accordingly, unlike the present invention's evaluation operation, Gafter's method is limited to algorithms with predictable size whose paths can be enumerated. Additionally, because the present invention passes the algorithm, this complication of Gafter is avoided.

The present invention's evaluation process occurs entirely through the node's evaluation process where the same external/internal agent requests value of a certain node's output and causes the node to evaluate its output and consequently its input. See page 7, lines 3-5; page 6, lines 21-28. Thus, the evaluation process of the present invention does not necessarily proceed in succession but depends on the dynamic changes in the control flow as indicated by the algorithms in each node. See independent claim 22 of the present invention. Therefore, the present invention provides an operation that prevents reevaluation of certain nodes which do not need to be evaluated. See page 2, lines 16-22.

The Examiner also stated that the present invention's self-evaluating data structure included in the function that is passed to the node is anticipated by Gafter's computation of reverse dominators from the CFG. According to Gafter, reverse dominators of a given node are defined as the set of nodes through which control from that given node is guaranteed to pass. See column 3, line 66 through column 4, line 1. For example, for the software code discussed previously, after the CFG is created, then the reverse dominators that list the guarantee path through the respective nodes are determined. See column 4, lines 20-25. The present invention's self-evaluating data structure does not need to perform such a determination because the variation of different outcomes are not predetermined but are performed dynamically, which is allowed by the algorithm passing operation. See page 6, lines 13 through page 7 line 2; independent claim 17. Accordingly, due to the dynamic evaluation process in the dependency graph, there need not be any reverse dominators. And even if there were reverse dominators, they could change between evaluations and are not predefined as in Gafter.

Accordingly, independent claims 1, 17,19, and 20-22 and claims depending therefrom are not anticipated by Gafter, thus, withdrawal of the rejection is respectfully requested.

**NEW CLAIMS:**

New claim 23 has been added to highlight an aspect of the invention that enables "passing an algorithm associated with a first dependency node" for reexecution of the algorithm "as part of an evaluation of the second node" each time new input data is entered. As a result, the present invention avoids the task of creating many different nodes for each variation outcome of an algorithm based on different input data.

New claim 24 has been added to emphasize the method of "passing an algorithm" between dependency nodes in a graphics creation process allowing the receiving dependency node to execute the received algorithm "as part of an evaluation" of the receiving node. Accordingly, unlike the Gafter CFG, when a particular attribute is evaluated, the result is not the single value of that attribute with the current state of inputs, but instead an algorithm that can be used to evaluate the attribute at any desired combinations of inputs or graph states.

Further, new claim 25 has been added to clarify that unlike Gafter, another aspect of the present invention discloses a method of evaluating a dependency graph by "passing an algorithm" where the algorithm comprises an algorithm calling technique allowing the "calling" of the passed algorithm "via the second dependency node" to reexecute the algorithm each time input data changes.

Gafter does not anticipate the methods of evaluation of a dependency graph of the present invention because Grafter's method involves pre-translation of each varied outcome of the algorithm into a CFG and is incapable of dynamically identifying the parameters of the algorithm.

**CONCLUSION:**

In accordance with the foregoing, it is respectfully submitted that all outstanding objections and rejections have been overcome and further, that all pending claims are not anticipated. Thus, there being no further outstanding objections or rejections, the application is submitted as being in condition for allowance, which action is earnestly solicited.

Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 12/10/03

By: Richard A. Gollhofer  
Richard A. Gollhofer  
Registration No. 31,106

1201 New York Avenue, NW, Suite 700  
Washington, D.C. 20005  
Telephone: (202) 434-1500  
Facsimile: (202) 434-1501